A Parallel Software Development for Watershed Simulations

Jing-Ru C. Cheng[1], Robert M. Hunter[1], Hwai-Ping Cheng[2], and David R. Richards[3]

[1] Major Shared Resource Center,
Information Technology Laboratory
{ruth.c.cheng, robert.m.hunter}@erdc.usace.army.mil
[2] Coastal and Hydraulics Laboratory
hwai-ping.cheng@erdc.usace.army.mil
[3] Information Technology Laboratory,
U.S. Army Engineer Research and Development Center,
Vicksburg, MS 39180-6199, USA
david.r.richards@erdc.usace.army.mil

Abstract. A watershed software application is designed to model a coupled system of multiple physics on multiple domains. Tremendous computational resources are required to integrate the system equations on large spatial domains with multiple temporal scales among them. Supported by the Department of Defense Common High Performance Computing Software Initiative, the parallel WASH123D software development aims to efficiently simulate one aspect (i.e., soil and land) of the battlespace environment. Currently, the coupled two-dimensional overland and three-dimensional subsurface flows have been completed. Different numerical approaches are implemented to solve different components of the coupled system. The parallelization of such a complex system is developed on an IT-based approach—modular, hierarchical model construction, portable, scalable, and embedded parallel computational tools development and integration. Experimental results are presented to demonstrate the successful implementation of the parallel algorithms. Detailed profiling is also provided to show the imposed light-weight communication overhead.

1 Introduction

Watershed models simulate major hydrological processes on multiple spatial domains over varied temporal scales with interactions among them spanning from uncoupled to strongly coupled. Different numerical approaches for such a coupled nonlinear hydrologic process have been proposed to be efficient and affordable. Penn State Integrated Hydrologic Model (PIHM) [1] integrates hydrological models using the so-called "semi-discrete" method, which reduces the governing partial differential equations (PDE) to ordinary differential equations (ODE), while Yeh et al. [2] presented a first-principle, physics-based watershed model. According to Yeh's review [3], HSPF (Hydrologic Simulation Program—FORTRAN) and WASH123D are the only models that include complete media systems, i.e., stream/rivers, overland regimes, and subsurface media, and encompass the complete suite of fluid flows and thermal, salinity, sediment, and chemical transport processes. The difference between them is that HSPF employs the parametric approach, while WASH123D is based on a first-principle, physics-based approach. The drawback of the parametric approach is explained in Sect. 2.3.

The U.S. Army Corps of Engineers plays a critical role in the Nation's watershed management. In addition, supported by the Department of Defense (DoD) Common High Performance Computing Software Initiative (CHSSI), the parallel WASH123D software development aims to efficiently simulate one aspect of the battlespace environment, which includes space, weather, ocean, and soil, towards development of a complete battlespace environment. The current accomplishment, including the development of the numerical approaches, software designs, and parallel algorithms, is presented in this paper.

2 Numerical Algorithms in WASH123D

The governing equations of two-dimensional (2-D) overland flow and three-dimensional (3-D) subsurface flow, and the numerical approaches solving the system equations are described in the following subsections. The numerical methods that this paper presents are those used for demonstration in Sect. 4.

2.1 2-D Overland Flow

The 2-D overland flow is computed by solving the depth-averaged diffusive wave equation with the semi-Lagrangian finite element method (FEM). The governing equation can be written as

$$\frac{\partial h}{\partial t} + \nabla \cdot \mathbf{q} = S + R - E - I \quad \text{or} \quad \frac{\partial h}{\partial t} + \nabla \cdot [\mathbf{V}h] = S + R - E - I \ , \tag{1}$$

where h = overland water depth[L], t = time[t], $\mathbf{q}$ = overland flux[L^3/t/L], S = man-induced source[L^3/t/L^2], R = rainfall rate[L/t], E = evapotranspiration rate[L/t], I = infiltration rate [L/t], and $\mathbf{V}$ = overland flow velocity[L/t]. With the semi-Lagrangian FEM, (1) can be written in Lagrangian form, then integration along its characteristic line yields

$$\left(1 + \frac{\triangle\tau}{2} K_i{}^{(n+1)}\right) h_i{}^{(n+1)} = \left(1 - \frac{\triangle\tau}{2} K_i{}^{*}\right) h_i{}^{*} + \frac{\triangle\tau}{2}\left(S_i{}^{(n+1)} + S_i{}^{*}\right) + \frac{\triangle\tau}{2}\left(R_i{}^{(n+1)} + R_i{}^{*}\right) - \frac{\triangle\tau}{2}\left(E_i{}^{(n+1)} + E_i{}^{*}\right) - \frac{\triangle\tau}{2}\left(I_i{}^{(n+1)} + I_i{}^{*}\right) \ , \tag{2}$$

where $K = \nabla \cdot \mathbf{V}$, $\triangle\tau$ = the tracking time[t], which equals $\triangle t$ (the time interval) when the backward tracking is carried out all the way to the root of the characteristic line, but is less than $\triangle t$ when the backward tracking hits the boundary before $\triangle t$ is completely consumed; ${K_i}^{(n+1)}$, ${h_i}^{(n+1)}$, ${S_i}^{(n+1)}$, ${R_i}^{(n+1)}$, ${E_i}^{(n+1)}$, and ${I_i}^{(n+1)}$ are the values of K, h, S, R, E, and I, respectively, at $\mathbf{x}_i$ at new time $t = (n+1)\triangle t$; and ${K_i}^*$, ${h_i}^*$, ${S_i}^*$, ${R_i}^*$, ${E_i}^*$, and ${I_i}^*$ are the values at ${\mathbf{x}_i}^*$, i.e., where the backward tracking ends. Equation (2) is used to compute the water depth, h, at all nodes except for the upstream boundary nodes, where water depth is determined by applying adequate boundary conditions. Since the flow velocity can be computed and represented as a function of water depth, (2) is used in a nonlinear iteration loop until a convergent solution is obtained.

2.2 3-D Subsurface Flow

The governing equation of subsurface flow through saturated-unsaturated porous media can be derived based on the conservation law of water mass and can be written as follows.

$$\frac{d\theta}{dh}\frac{\partial h}{\partial t} = \nabla \cdot [\mathbf{K} \cdot (\nabla h + \nabla z)] + q \ , \tag{3}$$

where θ = moisture content[L^3/L^3], h = pressure head [L], $\mathbf{K}$ = the hydraulic conductivity tensor[L/t], z = the potential head[L], and q = man-induced source [L^3/L^3/t]. Equation (3), the well-known Richards equation, is solved with the Galerkin FEM that can be found elsewhere [2].

2.3 2- and 3-D Coupling

The fluxes between surface and subsurface media are computed by imposing continuity of fluxes and state variables (e.g., overland water depth and subsurface pressure head). If the state variables exhibit discontinuity, then a linkage term is used to simulate the fluxes. Considering the interaction between the 2-D overland and 3-D subsurface flows, the pressures in the overland flow (if present) and in the subsurface media must be continuous across the interface. Thus, the interaction must be simulated by imposing continuity of pressures and fluxes as

$$h^o = h^s \quad \text{and} \quad Q^o = Q^s \implies I = \mathbf{n} \cdot \mathbf{K} \cdot (\nabla h^s + \nabla z) \ , \tag{4}$$

where h^o is the water depth[L] in the overland if it is present, h^s is the pressure head[L] in the subsurface, Q^o is the flux [L^3/L^2/t] from the overland to the interface, Q^s is the flux from the interface to the subsurface media [L^3/L^2/t], and $\mathbf{n}$ is an outward unit vector of the ground surface. The use of a linkage term such as $Q^o = Q^s = k(h^o - h^s)$, while convenient, is not appropriate because it introduces a nonphysical parameter, k. The calibration of k to match simulation with field data renders the coupled model *ad hoc* even though the overland and subsurface models are each individually physics-based.

Algorithm 2.1 depicts the 2-D/3-D coupling algorithm used in WASH123D. Ideally, overland flow and subsurface flow should be strongly coupled within each time-step. However, this would introduce unaffordable computational characteristics because small time intervals may be required for solving nonlinear 2-D overland flow when a high-resolution mesh is employed. To make computation affordable, in WASH123D each 3-D flow-time interval may contain more than one 2-D flow-time interval. The fluxes through the surface-subsurface interface are updated using (4) for 2-D/3-D in each 3-D coupling/nonlinear iteration.

Algorithm 2.1 The 2-D/3-D Coupling Algorithm in WASH123D

```
Foreach 3D flow time step ($\triangle t_{3DF}$) do
    Foreach 3D coupling/nonlinear iteration do
        Foreach 2D flow time step ($\triangle t_{2DF}$) do
            Incorporate infiltration/seepage for 2D/3D coupling
            Foreach 2D coupling/nonlinear iteration do
                Solve linearized 2D flow equation
            Endfor
        Endfor
        Incorporate infiltration/seepage for 2D/3D coupling
        Solve linearized 3D flow equation
    Endfor
Endfor
```

3 Parallel Algorithms and Implementation

In WASH123D, different numerical approaches are implemented to solve different components of the coupled system. The parallelization of such a complex model starts with the data structure design and then tackles the programming paradigm. The original serial computational kernel is included in the parallel software without any changes to shorten the development time, because there is no parallelization involved. Therefore, the data structure design becomes very important if the goals of object orientation, parallelization, software integration, and language interoperability are to be reached.

3.1 Data Structure Design

To account for problem domains that may include 1-D river/stream network, 2-D overland regime, and 3-D subsurface media, three `WashMesh` objects are constructed in the object `WashDomain`, which embraces the computational domain as sketched in Fig. 1. These three objects describe the three subdomains, on which a set of governing equations is derived to mathematically describe the behavior of the component within the entire domain. Note that the `WashDomain` also includes a coupling object named `WashCouple`. Moreover, the `WashGlobal` object describes the common phenomena, and the `WashProcinfo` object sets up the parallel environment context. Each subdomain (i.e., `WashMesh`) is partitioned, based on users' partitioning criteria, to processors by DBuilder [4]. Hence, each `WashMesh` object may include `vtxDomain` and `elementDomain`, which are created and managed by DBuilder, to maintain coherent data structures among processors via ghost vertices/elements on a given mesh. The `WashCouple` object may include the coupler for (1-D, 2-D), (1-D, 3-D), and/or (2-D, 3-D) interactions. The coupler encapsulates all the implementation of a Message Passing Interface (MPI) scheme for communication/synchronization between different

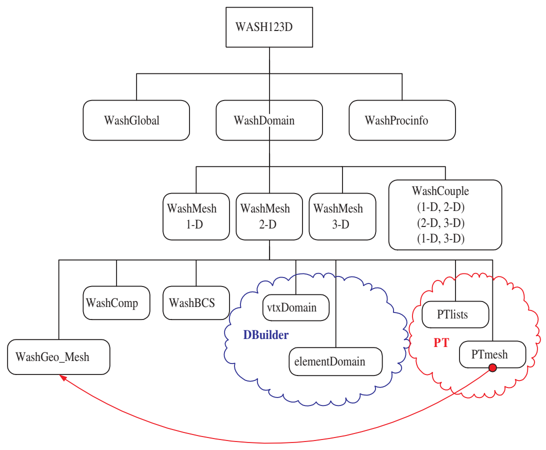

Fig. 1. Data structure design of the parallel WASH123D

`WashMesh` objects. This approach can partition each subdomain (i.e., `WashMesh`) independently. Therefore, this software tool can be reused to integrate two or more applications with different physics on multidomains.

3.2 Parallel Particle Tracking Software Integration

To solve the 2-D overland flow problem, with the Picard method solving the nonlinearity of the 2-D overland flow (2), the linearized equation can be solved by using particle tracking to compute the total time-derivative term in (1) and by manipulating the integration along the tracking path for the source/sink terms. Naturally, this application is perfectly suited for parallel implementation because the dependent variable, either water depth or water stage, can be obtained by solving the linearized equation independently. For such a purpose, the parallel particle tracking (PT) software [5] is facilitated with a new pathline computation kernel to accurately track particles under unsteady flow fields [6]. The design goal of the PT software development is to interface with different parallel or mesh programming environments. This goal is achieved through a software architecture specifying a lightweight functional interface [5].

3.3 DBuilder Software Toolkit Development

DBuilder [4] provides support of domain partitioning, parallel data management, coupling coordination, and parallel solver interface. This toolkit embeds all the MPI function calls required by the application codes and provides a set of user-interface functions to retrieve/modify parallel data. The basic concept of the development is based on a hierarchical modular design. To meet the requirements of different numerical methods, DBuilder can build a vertex domain with a distributed number of vertices, an element domain with a distributed number of elements, and a boundary element domain comprised of boundary elements in the element domain. DBuilder provides a default rule and a callback approach to users for the coordination between vertices and elements required by finite element applications.

The synchronization mechanism in DBuilder includes two steps. First, the local data that are shared on other processors are packed into a contiguous memory section, i.e., data are packed sequentially based on the receiving processor's rank. Within this set the data are then ordered based on global index values. Once the data have been packed, a single call to `MPI_Alltoallv` is made to update the ghost data on all processes. The computational cost of the pack routine for a given process is O(N), where N is the number of local vertices. The low computational cost is achieved by construction and storage of a map of local data indices to the packed array indices. The communication cost of `MPI_Alltoallv` has a worst cost of $O(N^2)$, where N is the number of processors.

Multiphysics applications on multidomains have become a large focus. This requires executing multidomain integration of two or more applications. The spatial relationship between computational domains can be adjacent, partially/fully overlapped, or distinct. DBuilder allows for the building of a coupler object to avoid the dependency between meshes when partitioning. This functionality may extend the reuse of this software toolkit to coupling different areas of applications, e.g., ocean and atmosphere coupled systems. Details can be found in [4].

4 Experimental Results

Fort Benning military base is located in the Upatoi River watershed, west-central Georgia south of the city of Columbus, GA, and east of Phenix City, AL. Fort Benning is a major training area for the U.S. Army Infantry. The 2-D overland domain, which covers about 450 square miles, is discretized with 103,619 vertices and 206,167 elements. The underlying 3-D domain contains 3,092,505 elements and 1,657,904 vertices.

Figure 2 plots the wall clock time vs. number of processors for the coupled 2-D overland flow simulation and 3-D subsurface flow simulation on the Fort Benning site on three different architectures. At the Engineer Research and Development Center Major Shared Resource Center (ERDC MSRC), the Compaq AlphaServer SC45 machine is configured with 128 nodes. Each node has four 1-GHz processors connected by a 64-port, single-rail Quadrics high-speed interconnect switch. At the Naval Oceanographic Office MSRC, the IBM P4 machine has 168 nodes. Each node has eight 1.3-GHz CPUs and 8 GBs of memory. Nodes communicate through the IBM's Colony II switch. At the Army Research Laboratory (ARL) MSRC, the linux cluster, Evolocity II, has 256 processors. Each has a 3.06-GHz CPU using Myrinet interconnect for communication. Figure 3 shows the communication overhead taken up in the total wall clock time on these three different machine architectures.

From these figures, one can observe that ARL's linux cluster outperforms the others except when the simulation runs on 32 processors. For the 64-processor simulation, the parallel efficiency is around 74 percent on the Evolocity II linux

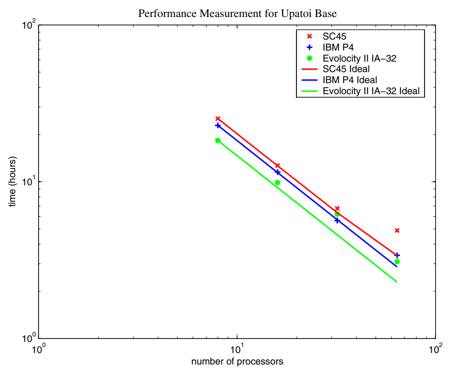

Fig. 2. Performance measurement on three machines

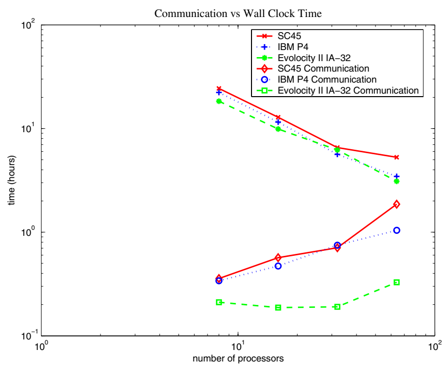

Fig. 3. Communication overhead compared with the total wall clock time

Table 1. Scalability on three machines

	Compaq SC45			IBM P4			Evolocity II Linux cluster		
nproc	Time (sec)	Speedup	PE	Time (sec)	Speedup	PE	Time (sec)	Speedup	PE
8	90978.28	—	—	82379.64	—	—	65983.72	—	—
16	45677.70	15.934	0.995	41497.16	15.882	0.993	35597.69	14.829	0.927
32	24316.91	29.931	0.939	20326.74	32.422	1.013	22360.68	23.608	0.738
64	17590.44	41.376	0.691	12224.27	53.912	0.842	11143.44	47.370	0.740

cluster, listed in Table 1, around 84 percent on the IBM P4 (see Table 1), and less than 70 percent on the SC45 (see Table 1). The main cause for such a difference is that the IBM P4 has better scalability with communication (see Fig. 3) and the Evolocity II has better CPU speed.

Figure 4 plots the high-water memory marks to show the memory scalability. While the parallel implementation imposes memory overhead, the memory requirements per processor are greatly reduced. The reduction of memory can then reduce the cache miss or page swapping, which can improve the performance significantly.

5 Summary and Future Plans

The software tool DBuilder has successfully embedded MPI routines so that application developers do not need to know the MPI library and parallel algorithms. The result shows that the implementation in DBuilder has successfully partitioned the domain, balanced the workload, and scaled the memory usage among processors. The following tasks for the software development have been completed: implementation of dynamic memory allocation, DBuilder functionality enrichment, parallel PT software integration, and the parallel performance

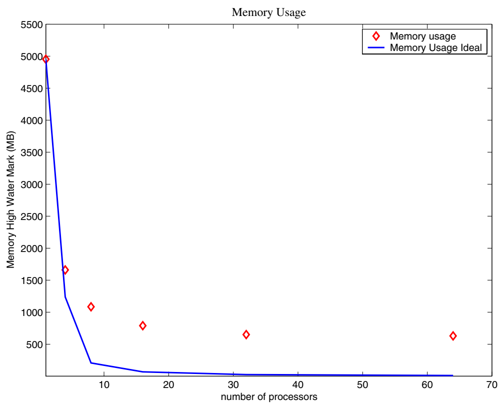

Fig. 4. Memory usage when run on different numbers of processors

evaluation. From the experimental demonstration, verification, fixed problem scalability, and memory scalability have been investigated. Further detailed profiling such as communication and memory overhead of each component will be performed. The WASH123D team has developed the 1-D module and is working on the 1-, 2-, and 3-D coupling module as well. Large-scale field problems are currently prepared for a large number of processors.

Acknowledgments

This work was supported in part by an allocation of computer time from the Department of Defense High Performance Computing Modernization Program at the U.S. Army Engineer Research and Development Center Major Shared Resource Center, Information Technology Laboratory, Vicksburg, Mississippi. The authors would like to thank their colleague Jerry Lin at the ERDC Coastal and Hydraulics Laboratory for preparing the data set for performance evaluation.

References

[1] Duffy, C.J.: Semi-discrete dynamical model for mountain-front recharge and water balance estimation, Rio Grande of southern Colorado and New Mexico. In: Groundwater Recharge in a Desert Environment: The Southwestern United States. Water Science and Applications Series, American Geophysical Union, D.C. (2004) 255–271

[2] Yeh, G.T., Cheng, H.P., Huang, G., Zhang, F., Lin, H.C., Edris, E., Richards, D.: A numerical model of flow, heat transfer, and salinity, sediment, and water quality transport in WAterSHed systems of 1-D stream-river network, 2-D overland regime, and 3-D subsurface media (WASH123D: Version 2.0). Technical Report Draft, Engineer Research and Development Center, U.S. Army Corps of Engineers, MS (2003)

[3] Yeh, G.T.: A rigorous treatment of interactions between various media in first principle, physics-based flow and transport modeling in watersheds. In: EOS Transac. Volume 83., American Geophysical Union, D.C. (2002) F436

[4] Hunter, R.M., Cheng, J.R.C.: DBuilder: A parallel data management toolkit for scientific applications. In: DoD High Performance Computing Modernization Program, 2004 Users Group Conference, Williamsburg, VA (June 7-11, 2004)

[5] Cheng, J.R.C., Jones, M.T., Plassmann, P.E.: A portable software architecture for mesh independent particle tracking algorithms. Parallel Algorithms and Applications (June-September 2004) 145–161

[6] Cheng, J.R.C., Plassmann, P.E.: Development of parallel particle tracking algorithms in large-scale unsteady flows. In: SIAM Conf. on Parallel Processing for Scientific Computing, San Francisco, CA (Feb. 25-27, 2004)